

TWO PROPOSALS FOR ROBUST PCA USING SEMIDEFINITE PROGRAMMING

MICHAEL MCCOY AND JOEL A. TROPP

ABSTRACT. The performance of principal component analysis (PCA) suffers badly in the presence of outliers. This paper proposes two novel approaches for robust PCA based on semidefinite programming. The first method, *maximum mean absolute deviation rounding* (MDR), seeks directions of large spread in the data while damping the effect of outliers. The second method produces a *low-leverage decomposition* (LLD) of the data that attempts to form a low-rank model for the data by separating out corrupted observations. This paper also presents efficient computational methods for solving these SDPs. Numerical experiments confirm the value of these new techniques.

1. INTRODUCTION

Principal component analysis (PCA), proposed in 1933 by Hotelling [23], is a common technique for summarizing high-dimensional data. Principal components are designed to identify directions in which the observations vary most. As a consequence, PCA is often used to reduce the dimension of the data.

Statistics based on variance, such as principal components, are highly sensitive to outliers [43]. The literature on robust statistics contains a wide variety of techniques that attempt to correct this shortcoming [25]. Unfortunately, many of these approaches are based on intractable optimization problems or lack a principled foundation.

Our focus in this work is to develop new formulations for robust PCA that can be solved efficiently using convex programming algorithms. Our first proposal, which we call *maximum mean absolute deviation rounding* (MDR), exchanges the variance in the definition of PCA with a function less sensitive to outliers known as the mean absolute deviation. Although this formulation leads to a non-convex optimization problem, we demonstrate that it is possible to approximate the optimum by relaxing to a semidefinite program and randomly rounding the solution. This method can be viewed as a specific instance of projection-pursuit PCA [26].

Our second proposal uses a different semidefinite program to split the input data into the sum of a low-leverage matrix and a matrix of corrupted observations. We refer to this dissection as a *low-leverage decomposition* (LLD) of the data. This method is similar in spirit to the rank-sparsity decomposition of Chandrasekaran et al. [7]. While preparing this manuscript, we learned of an independent investigation into this formulation of robust PCA by Xu et. al. [46, 47].

We describe algorithms that solve these semidefinite programs efficiently, and we provide numerical experiments that confirm the effectiveness of these new techniques. We begin with a brief overview of our proposals before laying out the details in Sections 2 and 3.

1.1. The Data Model. Suppose that we have a family $\{\mathbf{x}_i\}_{i=1}^n$ of n observations in p dimensions. We form an $n \times p$ data matrix \mathbf{X} whose rows are the observations. The observations are assumed to be centered; that is, $\frac{1}{n} \sum_i \mathbf{x}_i \approx \mathbf{0}$. While our methods do not explicitly require the data to be centered, this hypothesis allows us to interpret principal components as directions of high variance in the data. We discuss practical centering approaches in Section 5.

Date: December 15, 2010.

This work has been supported in part by ONR awards N00014-08-1-0883 and N00014-11-1-0025, AFOSR award FA9550-09-1-0643, and a Sloan Fellowship. This research was performed while the authors were in residence at IPAM. The authors can be contacted via email at {mccoy,jtropp}@acm.caltech.edu or postal mail at Computing & Mathematical Sciences, 1200 E. California Blvd., MC 305-16, California Inst. Technology, Pasadena, CA 91125.

1.2. Maximizing the Mean Absolute Deviation. Our first method is designed to mitigate a source of sensitivity in classical principal component analysis. The top principal component \mathbf{v}_{PCA} is defined as a direction of maximum variance in the data:

$$\mathbf{v}_{\text{PCA}} = \arg \max_{\|\mathbf{v}\|_2=1} \sum_{i=1}^n |\langle \mathbf{x}_i, \mathbf{v} \rangle|^2. \quad (1.1)$$

The squared inner products in (1.1) may lead to outsized influence of outlying points because squaring a large number results in a huge number, which can drag the principal component away from the bulk of the data. We can reduce this effect by replacing the squared inner product with a measure of spread that is less sensitive. We propose the use of the absolute value of the inner product:

$$\mathbf{v}_{\text{MD}} = \arg \max_{\|\mathbf{v}\|_2=1} \sum_{i=1}^n |\langle \mathbf{x}_i, \mathbf{v} \rangle|, \quad (1.2)$$

where we have added the subscript MD to indicate that we have exchanged the variance in equation (1.1) with a measure of spread known as the *mean absolute deviation* (MD) [25, p. 2].

This revision results in some complications. The formulation (1.1) is an eigenvector problem which can be solved efficiently. In contrast, it is NP-hard to compute \mathbf{v}_{MD} . Nevertheless, we develop an efficient randomized algorithm that provably computes an approximate solution to (1.2). We call this approach *maximum mean absolute deviation rounding* (MDR).

Our main result, Theorem 2.2, states that, for any failure probability $\delta > 0$ and loss factor $\varepsilon > 0$, our algorithm produces a unit-norm vector \mathbf{v}_{MDR} such that

$$\sum_{i=1}^n |\langle \mathbf{x}_i, \mathbf{v}_{\text{MDR}} \rangle| \geq \sqrt{\frac{2}{\pi}} (1 - \varepsilon) \max_{\|\mathbf{v}\|_2=1} \sum_{i=1}^n |\langle \mathbf{x}_i, \mathbf{v} \rangle|.$$

The algorithm requires that we solve one semidefinite program (SDP) whose size is polynomial in the number of observations. Since SDPs are solvable in polynomial time using interior-point methods, our algorithm is tractable in principle. In practice, solving SDPs can be daunting even for moderately sized input data—say, more than 100 observations. To address this issue, we detail a technique of Burer and Monteiro [4, 5] that can usually solve the SDP efficiently, and in Section 5 we provide some numerical evidence that this approach succeeds.

We find additional components by greedily restricting the data to a subspace perpendicular to the previous components and solving (1.2) again.

This proposal is not without precedent. A more general formulation appears in Huber’s book [24, p. 203], and it is now known as *projection-pursuit PCA* (PP-PCA) [26]. We provide further detail on PP-PCA in Section 2.2 and discuss the history of the method in 4.1.

1.3. A Low-Leverage Decomposition. Our second proposal stems from a different interpretation of classical principal component analysis. Instead of viewing classical principal components as directions of maximum variance, we can view them as an optimal low-rank model for the data [6]. Suppose \mathbf{P}_\star is a matrix that solves

$$\begin{aligned} & \text{minimize} && \|\mathbf{X} - \mathbf{P}\|_{\text{F}} \\ & \text{subject to} && \text{rank}(\mathbf{P}) = T. \end{aligned}$$

The dominant principal components of \mathbf{X} are given by the T right singular vectors of \mathbf{P}_\star corresponding with the nonzero singular values of \mathbf{P}_\star .

With real data, one is often faced with the situation where entire observations are corrupted. If this is the case, we would still like to recover a low-rank model. We can develop as natural formulation for identifying a low-rank model using the well-known rank sparsity [15] and group sparsity [37] heuristics. We propose to decompose the data matrix as $\mathbf{X} = \mathbf{P}_{\text{LLD}} + \mathbf{C}_{\text{LLD}}$ by solving the semidefinite program

$$\begin{aligned} & \text{minimize} && \sum_i \sigma_i(\mathbf{P}) + \gamma \sum_j \|\mathbf{c}_j\|_2 \\ & \text{subject to} && \mathbf{P} + \mathbf{C} = \mathbf{X}. \end{aligned} \quad (1.3)$$

We have written $\sigma_i(\mathbf{P})$ for the i th singular value of \mathbf{P} and \mathbf{c}_i for the i th row of \mathbf{C} .

We view the optimal matrix \mathbf{P}_{LLD} as a surrogate for the low-rank approximation to the uncorrupted data, and the optimal matrix \mathbf{C}_{LLD} as an approximation of the corrupted data. The formulation (1.3) has an interesting property even when \mathbf{P}_{LLD} is not low-rank or \mathbf{C}_{LLD} is

not row-sparse: \mathbf{P}_{LLD} is guaranteed to be a low-leverage set of observations in a sense we make precise in Section 3.1. As a result, we refer to $\mathbf{X} = \mathbf{P}_{\text{LLD}} + \mathbf{C}_{\text{LLD}}$ as a *low-leverage decomposition* (LLD) of the data. We define the dominant LLD components as the right singular vectors of \mathbf{P}_{LLD} .

This optimization problem is similar to the rank-sparsity decomposition problem proposed in [7]; see also [6]. We discuss these ideas at more length in Section 4. As this manuscript was being prepared, we learned of an independent investigation of the program (1.3) for robust PCA by Xu et. al. [46, 47] that provides conditions for recovery of the support of the corruption and the row-space of the uncorrupted observations.

1.4. Road map. Sections 2 and 3 describe our proposals in more detail, including theoretical guarantees and practical algorithms. Section 4 offers an overview of previous work on robust PCA, while Section 5 describes numerical experiments illustrating the performance of our methods in various settings. A technical appendix contains the proofs of supporting results.

1.5. Notation. We work exclusively with real numbers. The symbols \mathbb{P} and \mathbb{E} denote probability and expectation, respectively. We use ∂ to denote the subgradient map.

Bold capital letters denote matrices while bold lower-case letters denote vectors. We represent the i th row of a matrix \mathbf{A} by \mathbf{a}_i and the j th entry of a vector \mathbf{a} by a_j . The adjoint of a matrix \mathbf{A} is written \mathbf{A}^* . When referring to matrix elements, we sometimes use the notation $[\mathbf{A}]_{ij}$, and similarly for vectors we use $[\mathbf{a}]_i$.

We use the compact convention for the singular value decomposition (SVD) of a matrix: when \mathbf{A} is rank r , we write its SVD as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where \mathbf{U} and \mathbf{V} have orthonormal columns, and $\mathbf{\Sigma}$ is a non-singular diagonal matrix whose entries are positive and are arranged in weakly decreasing order. The notation $\mathbf{A} \succcurlyeq \mathbf{B}$ denotes that $\mathbf{A} - \mathbf{B}$ is positive semidefinite.

1.5.1. Norms. We denote the ℓ_p vector norm as $\|\mathbf{u}\|_p = (\sum_i |u_i|^p)^{1/p}$ for $1 \leq p < \infty$ and $\|\mathbf{u}\|_\infty = \max_i |u_i|$. The Frobenius norm of a matrix is defined by $\|\mathbf{A}\|_{\text{F}}^2 = \langle \mathbf{A}, \mathbf{A} \rangle$, where $\langle \cdot, \cdot \rangle$ represents the standard inner product. The Moore–Penrose pseudoinverse of a matrix \mathbf{A} is denoted \mathbf{A}^\dagger .

We define the ℓ_p to ℓ_q operator norm and its dual respectively by

$$\|\mathbf{A}\|_{p \rightarrow q} = \sup_{\|\mathbf{u}\|_p=1} \|\mathbf{A}\mathbf{u}\|_q, \quad \text{and} \quad \|\mathbf{B}\|_{p \rightarrow q}^* = \sup_{\|\mathbf{A}\|_{p \rightarrow q}=1} \langle \mathbf{B}, \mathbf{A} \rangle.$$

Table 1 describes some of the specific operator norms used in this work. We also use the norms $\|\mathbf{A}\|_{2 \rightarrow 1}$ and $\|\mathbf{A}\|_{\infty \rightarrow 1}$, which lack such simple descriptions; see Sections 2.3 and 2.4.

The operator norm of the adjoint satisfies $\|\mathbf{A}^*\|_{q^* \rightarrow p^*} = \|\mathbf{A}\|_{p \rightarrow q}$ where p and q satisfy the conjugacy relations $1/p + 1/p^* = 1$ and $1/q + 1/q^* = 1$ with the convention $1/\infty = 0$.

TABLE 1. Summary of the norms used in this work.

Norm	Description	Description of Dual
$\ \mathbf{A}\ _{2 \rightarrow 2}$	Maximum singular value of \mathbf{A}	Sum of the singular values of \mathbf{A}
$\ \mathbf{A}\ _{2 \rightarrow \infty}$	Maximum ℓ_2 row norm of \mathbf{A}	Sum of the ℓ_2 row norms of \mathbf{A}
$\ \mathbf{A}\ _{1 \rightarrow \infty}$	Maximum absolute entry of \mathbf{A}	Sum of the absolute entries of \mathbf{A}

2. MAXIMUM MEAN ABSOLUTE DEVIATION ROUNDING

Our first method is based on the classical interpretation of the top principal component as the direction of maximum empirical variance in multidimensional data. It has long been recognized that the variance is highly sensitive to outliers in the data [43]. The field of robust statistics has reacted by developing and analyzing robust measures of spread known as robust scales; see [25, Ch. 5] or [30, Sec. 2.5]. This literature describes a generic method for determining robust principal components by replacing the variance with a robust measure of scale. Li and Chen [26] published the first investigation of this under the name *projection-pursuit PCA* (PP-PCA). Our proposal is a specific instance of PP-PCA with the mean absolute deviation scale (2.1). We show

that this formulation is computationally intractable, but we develop an algorithm that provably approximates its solution. To our knowledge, this is the first rigorous algorithm for PP-PCA with a robust scale.

2.1. Scales. A *scale* is a function that measures the spread of one-dimensional data [25, Ch. 5]. By far, the most common scale is the empirical standard deviation, defined¹ as

$$\text{std}(\mathbf{y}) = \left(\sum_i y_i^2 \right)^{1/2} = \|\mathbf{y}\|_2,$$

where we assume the data \mathbf{y} is centered. Of course, the standard deviation is not the only way to measure the spread of the data. An alternative proposal [25, p. 2] is the *mean absolute deviation* (MD). For centered data \mathbf{y} , the MD scale is defined as

$$\text{MD}(\mathbf{y}) = \sum_i |y_i| = \|\mathbf{y}\|_1. \quad (2.1)$$

More generally, a scale is a function $S : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $S(\alpha\mathbf{y}) = |\alpha|S(\mathbf{y})$. Scales are typically chosen so that they are less sensitive to outliers than the standard deviation. The robust statistics literature focuses on scales that have a positive breakdown point: the value of the scale cannot be arbitrarily corrupted by nefariously chosen observations, so long as the fraction of bad observations in the entire data set is small. Although the mean absolute deviation has a breakdown point of zero, it exhibits more efficient behavior than the standard deviation under contaminated distributions [43].

2.1.1. Scales for multivariate data. We extend the definition of scales to multivariate data by considering the scale of the data in a given direction. The projection of the rows of \mathbf{X} onto the unit direction \mathbf{u} is given by the product $\mathbf{X}\mathbf{u}$. Note that if \mathbf{X} is centered in the sense of Section 1.1, then the projection $\mathbf{X}\mathbf{u}$ is also centered by linearity. We define the scale of \mathbf{X} in the direction \mathbf{u} to be the scale of the projected data $S(\mathbf{X}\mathbf{u})$.

As noted in [24], this definition is equivariant under an orthogonal change of basis: for any \mathbf{Q} with $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, the scale of \mathbf{X} in the direction \mathbf{u} is equal to the scale of $\mathbf{X}\mathbf{Q}^*$ in the direction $\mathbf{Q}\mathbf{u}$.

2.2. Projection-Pursuit PCA. Classically, the top principal component is defined as the direction where the empirical standard deviation in the data is largest:

$$\mathbf{v}_{\text{PCA}} = \arg \max_{\|\mathbf{v}\|_2=1} \text{std}(\mathbf{X}\mathbf{v}). \quad (2.2)$$

A natural approach for finding robust components is to replace the standard deviation in (2.2) with a robust scale $S(\cdot)$, so that the robust component is the direction of maximum robust scale

$$\mathbf{v}_{\text{PP}} = \arg \max_{\|\mathbf{v}\|_2=1} S(\mathbf{X}\mathbf{v}).$$

We define further robust components inductively by adding orthogonality constraints:

$$\mathbf{v}_{\text{PP}}^{(k)} = \arg \max_{\substack{\|\mathbf{v}\|_2=1 \\ \mathbf{v} \perp \mathbf{v}_{\text{PP}}^{(j)} \forall j < k}} S(\mathbf{X}\mathbf{v}). \quad (2.3)$$

This greedy method of constructing orthogonal components based on robust scales goes by the name projection-pursuit PCA. This scheme was originally proposed by Huber [24, p. 203], but was first studied in detail by Li and Chen [26]. PP-PCA reduces to PCA when the scale is given by the standard deviation due to the variational characterization of eigenvectors by Courant and Fischer.

To implement the PP-PCA method, one only needs a method that finds the first component. We discuss how to enforce the orthogonality constraints in Section 2.6.1.

¹One usually defines scales so that they are unbiased estimates of the sample standard deviation when the data is drawn from a normal distribution. We are more interested in the direction of maximal scale rather than the value, so we can safely ignore the normalization factor.

2.3. PP-PCA with the MD Scale is NP-Hard. Finding the top principal component is an eigenvector problem that amounts to computing the direction where the norm $\|\cdot\|_{2 \rightarrow 2}$ is achieved. Similarly, PP-PCA with the MD scale amounts to finding a vector that achieves an operator norm. Indeed, the problem $\mathbf{v}_{\text{MD}} = \arg \max_{\|\mathbf{v}\|_2=1} \|\mathbf{X}\mathbf{v}\|_1$ is equivalent to the problem

$$\text{find } \|\mathbf{v}_{\text{MD}}\|_2 = 1 \text{ such that } \|\mathbf{X}\mathbf{v}_{\text{MD}}\|_1 = \|\mathbf{X}\|_{2 \rightarrow 1}. \quad (2.4)$$

Unfortunately, exchanging the ℓ_2 norm for the ℓ_1 norm leads to an NP-hard computational problem. To see this, we require the following result, which we establish in the Appendix.

Fact 2.1. *For each matrix \mathbf{X} , the identity $\|\mathbf{X}\|_{2 \rightarrow 1}^2 = \|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1}$ holds.*

Rohn [39] shows that there exists a class of well-conditioned positive matrices \mathcal{M} such that the existence of a polynomial-time algorithm for accurately computing $\|\mathbf{M}\|_{\infty \rightarrow 1}$ for all $\mathbf{M} \in \mathcal{M}$ implies $\text{P} = \text{NP}$. Since we can factor positive matrices $\mathbf{M} = \mathbf{R}\mathbf{R}^*$ in polynomial time using, for example, a Cholesky factorization, the existence of an accurate polynomial-time algorithm that computes $\|\mathbf{R}\|_{2 \rightarrow 1}^2$ for any matrix \mathbf{R} implies that $\text{P} = \text{NP}$.

The observation that Equation (2.3) is NP-hard to solve for the specific choice $S(\cdot) = \|\cdot\|_1$ has serious implications for existing PP-PCA algorithms. The algorithms available in the literature for PP-PCA [9, 11, 26] are general schemes that claim to work for any choice of scale S . As a result, none of these algorithms can provide both accurate and efficient solutions to the PP-PCA problem. This issue is not merely theoretical because these algorithms tend to perform poorly in practice. We discuss this point further in Section 4.1.

2.4. Approximating the $\ell_2 \rightarrow \ell_1$ Norm using Randomized Rounding. Although it is NP-hard to compute the $\ell_2 \rightarrow \ell_1$ norm, it is possible to approximate its value efficiently. This fact is a consequence of the little Grothendieck theorem [36, Sec. 5b], but the algorithm depends on ideas of Nesterov [34], a technique of Burer and Monteiro [4, 5], and a new factorization step.

2.4.1. The semidefinite relaxation of the $\ell_2 \rightarrow \ell_1$ norm. Before describing our algorithm, we begin by showing how the computation of $\ell_2 \rightarrow \ell_1$ operator norm can be relaxed to a semidefinite program. First, apply Fact 2.1 to change the computation of the $\ell_2 \rightarrow \ell_1$ norm to the computation of the $\ell_\infty \rightarrow \ell_1$ norm:

$$\|\mathbf{X}\|_{2 \rightarrow 1}^2 = \|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1} = \max_{\|\mathbf{y}\|_\infty=1} \mathbf{y}^* \mathbf{X}\mathbf{X}^* \mathbf{y}. \quad (2.5)$$

The second identity above follows from the proof of Fact 2.1; see also [39, Prop. 1]. Interpreting the quadratic form on the right hand side of (2.5) as a trace implies that $\|\mathbf{X}\|_{2 \rightarrow 1}^2$ is the optimal value of the (non-convex) program

$$\begin{aligned} & \text{maximize} && \text{trace}(\mathbf{X}\mathbf{X}^* \mathbf{Z}) \\ & \text{subject to} && \mathbf{Z} = \mathbf{y}\mathbf{y}^*, \quad [\mathbf{Z}]_{ii} = 1 \text{ for all } i. \end{aligned} \quad (2.6)$$

Relaxing the rank one constraint $\mathbf{Z} = \mathbf{y}\mathbf{y}^*$ to a positive-semidefinite constraint $\mathbf{Z} \succcurlyeq \mathbf{0}$ leads to the SDP

$$\begin{aligned} & \text{maximize} && \text{trace}(\mathbf{X}\mathbf{X}^* \mathbf{Z}) \\ & \text{subject to} && \mathbf{Z} \succcurlyeq \mathbf{0}, \quad [\mathbf{Z}]_{ii} = 1 \text{ for all } i. \end{aligned} \quad (2.7)$$

It follows that $\|\mathbf{X}\|_{2 \rightarrow 1} \leq \alpha_\star$, where α_\star^2 is the optimal value of (2.7). Moreover, Grothendieck's inequality for positive-semidefinite matrices implies that

$$\alpha_\star^2 \leq \frac{\pi}{2} \|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1}, \quad (2.8)$$

where this inequality is asymptotically the best possible [2, Sec. 4.2]. Thus, α_\star is within a factor of $\sqrt{\pi/2} < 1.26$ of the true value of the norm $\|\mathbf{X}\|_{2 \rightarrow 1}$.

Algorithm 1: Maximum Mean Absolute Deviation RoundingINPUT: An $n \times p$ matrix \mathbf{X} ; repetition count K .OUTPUT: A $p \times 1$ unit-norm vector \mathbf{v}_\star and an optimal value α_\star .

- (1) Find an
- \mathbf{R}_\star
- such that
- $\mathbf{Z}_\star = \mathbf{R}_\star \mathbf{R}_\star^*$
- solves the semidefinite program

$$\begin{aligned} & \text{maximize} && \text{trace}(\mathbf{Z} \mathbf{X} \mathbf{X}^*) \\ & \text{subject to} && \mathbf{Z} \succcurlyeq \mathbf{0}, \quad [\mathbf{Z}]_{ii} = 1 \text{ for } i = 1, \dots, n \end{aligned} \quad (2.10)$$

Set α_\star to be the square root of the optimal value: $\alpha_\star = \sqrt{\text{trace}(\mathbf{Z}_\star \mathbf{X} \mathbf{X}^*)}$.

- (2) For each
- $k = 1, \dots, K$
- , do

(a) Set $\mathbf{y}^{(k)} = \text{sgn}(\mathbf{R}_\star \mathbf{g}^{(k)})$, where $\mathbf{g}^{(k)}$ is an $n \times 1$ standard normal random vector.(b) Set $\mathbf{v}^{(k)} = \mathbf{X}^* \mathbf{y}^{(k)} / \|\mathbf{X}^* \mathbf{y}^{(k)}\|_2$.

- (3) Set
- $\mathbf{v}_\star = \arg \max_{k=1, \dots, K} \|\mathbf{X} \mathbf{v}^{(k)}\|_1$
- .

2.5. The MDR Algorithm. The fact that equation (2.7) gives us a good upper bound on the value of $\|\mathbf{X}\|_{2 \rightarrow 1}$ is of secondary importance. We would prefer an approximation for \mathbf{v}_{MD} in (2.4), that is, a vector \mathbf{v}_\star with $\|\mathbf{v}_\star\|_2 = 1$ such that $\|\mathbf{X} \mathbf{v}_\star\| \approx \|\mathbf{X}\|_{2 \rightarrow 1}$. We accomplish this goal via a randomized procedure that rounds an optimal solution \mathbf{Z}_\star to (2.7) back to a vector \mathbf{v}_\star . The entire procedure is detailed in Algorithm 1.

The first step of the algorithm solves the SDP relaxation (2.7). In Step 2(a), we draw a random $\mathbf{y} \in \{\pm 1\}^n$ with $\mathbb{E} \|\mathbf{X} \mathbf{X}^* \mathbf{y}\|_1 = 2\alpha_\star^2/\pi$. This procedure is well understood [34]. The method in Step 2(b) that we use to compute \mathbf{v} from \mathbf{y} is novel, and it requires a proof of correctness, which appears in the Appendix. By choosing the best random outcome, Step 3 limits the probability that our method fails to provide a reasonable approximation.

The following theorem describes the behavior of Algorithm 1.

Theorem 2.2. *Suppose that \mathbf{X} is an $n \times p$ matrix, and let K be the number of rounding trials. Let $(\mathbf{v}_\star, \alpha_\star)$ be the output of Algorithm 1. Then $\alpha_\star \geq \|\mathbf{X}\|_{2 \rightarrow 1}$. Moreover, for $\theta < 1$, the inequality*

$$\|\mathbf{X} \mathbf{v}_\star\|_1 > \theta \sqrt{\frac{2}{\pi}} \alpha_\star \quad (2.9)$$

holds except with probability $e^{-2K(1-\theta^2)/\pi}$.

In Theorem 2.2, it may be more natural to specify a failure probability $\delta > 0$ and approximation loss $\varepsilon = 1 - \theta > 0$ instead of a repetition number K . In this case, simple algebra shows that $\|\mathbf{X} \mathbf{v}_\star\|_1 > (1 - \varepsilon) \sqrt{2/\pi} \|\mathbf{X}\|_{2 \rightarrow 1}$ except with probability δ , so long as

$$K \geq \frac{\pi}{2} \cdot \frac{\log(1/\delta)}{\varepsilon(2 - \varepsilon)} = \mathcal{O}(\varepsilon^{-1} \log(\delta^{-1})).$$

In particular, the choice $K = 94$ implies that $\|\mathbf{X} \mathbf{v}_\star\|_1 > 0.75 \|\mathbf{X}\|_{2 \rightarrow 1}$ with probability at least 0.999.

We use the approximation ratio $\rho = \|\mathbf{X} \mathbf{v}_\star\|_1 / \alpha_\star$ to measure the quality of the optimal solution in Section 5. Although Theorem 2.2 only guarantees that we can make ρ as close to $\sqrt{2/\pi} > 0.79$ as we desire, in practice we typically see a 0.95 approximation ratio or higher. This observation does not indicate that the analysis of the algorithm is loose; it follows directly from [2, Sec. 4.2] that this bound is asymptotically tight for a class of examples as $n \rightarrow \infty$.

2.6. Implementation of Algorithm 1. For a fixed iteration count K , the complexity of Algorithm 1 is typically dominated by Step 1. When applied to (2.10), modern interior-point methods are guaranteed to compute the optimal objective value α_\star and optimal point \mathbf{Z}_\star accurately in polynomial time. The factor \mathbf{R}_\star is determined using a Cholesky factorization of \mathbf{Z}_\star . In practice, interior-point methods are very slow for large-scale problems, so we prefer an algorithm of Burer and Monteiro [5].

The algorithm of Burer and Monteiro never forms the semidefinite matrix \mathbf{Z} ; rather it operates directly with the factor \mathbf{R} . We express the objective function of (2.10) in terms of \mathbf{R} as

$\text{trace}(\mathbf{R}\mathbf{R}^*\mathbf{X}\mathbf{X}^*) = \|\mathbf{X}^*\mathbf{R}\|_{\text{F}}^2$. The constraints $[\mathbf{Z}]_{ii} = 1$ are equivalent to constraints on the rows of \mathbf{R} of the form $\|\mathbf{r}_i\|_2 = 1$.

We implicitly enforce these row constraints by incorporating them into the objective function as in [4, Sec. 4.2]. The resulting unconstrained, nonconvex optimization problem takes the form

$$\text{maximize}_{\mathbf{R}} \|\mathbf{X}^*\mathcal{N}(\mathbf{R})\|_{\text{F}}^2, \quad (2.11)$$

where $\mathcal{N}(\mathbf{R})$ denotes the operator that normalizes the rows of \mathbf{R} , that is, $[\mathcal{N}(\mathbf{R})]_{ij} = [\mathbf{r}_i]_j / \|\mathbf{r}_i\|_2$.

We then apply a conjugate gradient algorithm to maximize the unconstrained objective in (2.11). Our particular implementation uses the algorithm of Hager and Zhang [22], which we have found to work well in our experiments. We refer to our online code for the choice of parameters in this conjugate gradient algorithm [31].

This factorization technique for solving (2.10) is advantageous because it reduces the dimension of the problem. The paper [5] shows that restricting \mathbf{R} to be an $n \times k$ matrix for $k = \mathcal{O}(\sqrt{n})$ suffices to solve this problem exactly. To be precise, when $k = \lfloor (1 + \sqrt{9 + 8n})/2 \rfloor$ any *local* minimum $\mathbf{R}_* \in \mathbb{R}^{n \times k}$ of (2.11) gives a *global* minimum \mathbf{Z}_* of (2.10) via the map $\mathbf{Z}_* = \mathbf{R}_*\mathbf{R}_*^*$, provided a mild technical condition² holds.

2.6.1. Orthogonal Restriction. Algorithm 1 only approximates the first principal component in (1.2). In order to approximate the k th robust principal component for $k > 1$, we define a new matrix \mathbf{X}_k by restricting the rows of \mathbf{X} to the subspace perpendicular to the span of $\mathbf{v}_1, \dots, \mathbf{v}_{k-1}$. Ignoring numerical stability, we can inductively define

$$\mathbf{X}_k = \mathbf{X}_{k-1} - \mathbf{X}\mathbf{v}_{k-1}\mathbf{v}_{k-1}^* = \mathbf{X}\left(\mathbf{I} - \sum_{j=1}^{k-1} \mathbf{v}_j\mathbf{v}_j^*\right), \quad (2.12)$$

which ensures each row of \mathbf{X} is orthogonal to the previous components \mathbf{v}_j for $j < k$. We then apply Algorithm 1 to the restricted matrix \mathbf{X}_k to produce the component \mathbf{v}_k . Since the output \mathbf{v}_* of Algorithm 1 is a linear combination of the rows of the input matrix by Step 2(b), this iterative procedure ensures that \mathbf{v}_k is perpendicular to the previous components.

In practice, the implementation can be done using Householder reflections as in [11]; see [42] for further background on the implementation of Householder transformations. Householder reflections are more numerically stable than the naïve method (2.12). Moreover, they take full advantage of the fact that we are only searching over a $p - k + 1$ dimensional subspace by reducing the dimension of \mathbf{X}_k to $n \times (p - k + 1)$.

2.7. Extending the Rounding to Multiple Components. We have also attempted to extract a collection of robust components simultaneously by solving a single semidefinite program. That is, we would like to solve the problem

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^T \|\mathbf{X}\mathbf{v}_i\|_1 \\ &\text{subject to} && \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij} \end{aligned} \quad (2.13)$$

where δ_{ij} is the Kronecker delta function. When $T = 1$, equation (2.13) is equivalent with (1.2). When $T > 1$, the restriction $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$ ensures that the optimum occurs at an orthogonal set of unit vectors.

We can rephrase this optimization problem by the equivalent quadratically constrained quadratic program

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^n \mathbf{w}_i^* \mathbf{X} \mathbf{v}_i \\ &\text{subject to} && \text{diag}(\mathbf{w}_i \mathbf{w}_i^*) = 1, \quad \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij} \end{aligned} \quad (2.14)$$

The diagonal restrictions on \mathbf{w}_i ensure that $\mathbf{w}_i \in \{\pm 1\}^n$ for each $i = 1, \dots, n$. The nonconvex problem (2.14) can be approximated via a semidefinite relaxation proposed in [33]. The results of [41] imply that the optimal value of this relaxation is guaranteed to be larger than the optimal value of (2.13) by no more than a logarithmic factor. The rounding procedure does not produce orthogonal vectors, so we need to apply an additional orthogonalization step to achieve feasibility for (2.13). Empirically, we have found that the orthogonalization increases the objective value over the standard rounding, so it appears that there is no loss in applying this procedure.

²Specifically, the objective function $\text{trace}(\mathbf{Z}\mathbf{X}\mathbf{X}^*)$ must not be constant along a face of the feasible set.

Algorithm 2: Low-Leverage DecompositionINPUT: An $n \times p$ data matrix \mathbf{X} ; desired number of principal components T .OUTPUT: A $p \times T$ matrix \mathbf{V}_\star with orthogonal columns.

- (1) Find
- $(\mathbf{P}_\star, \mathbf{C}_\star)$
- that solve

$$\begin{aligned} & \underset{(\mathbf{P}, \mathbf{C})}{\text{minimize}} && \|\mathbf{P}\|_{2 \rightarrow 2}^* + \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^* \\ & \text{subject to} && \mathbf{P} + \mathbf{C} = \mathbf{X} \end{aligned} \quad (3.2)$$

- (2) Compute the SVD
- $\mathbf{P}_\star = \mathbf{U}\Sigma\mathbf{V}^*$
- .

- (3) Set
- \mathbf{V}_\star
- to the first
- T
- columns of
- \mathbf{V}
- , that is, set

$$[\mathbf{V}_\star]_{ij} = [\mathbf{V}]_{ij} \text{ for } i = 1, \dots, p, \text{ and } j = 1, \dots, T.$$

Unfortunately, this method does not appear to be competitive with the projection pursuit method. The vectors we find by coupling Algorithm 1 with the orthogonal pursuit of Section 2.6.1 are feasible for (2.14) and typically provide a larger objective value than rounding coupled with post-processing orthogonalization. A better rounding procedure for this type of relaxation may prove more effective than the projection-pursuit approach; this is a direction for further research.

3. THE LOW-LEVERAGE DECOMPOSITION

Our second method is derived from the interpretation of principal component analysis as a matrix approximation problem. When the observations are drawn from a highly correlated family, the singular values of the data matrix \mathbf{X} tend to decay rapidly. If this is the case, then the matrix \mathbf{X} is well approximated by a low-rank matrix \mathbf{P} .

It is rare that a large data set can be compiled without error, but it is often the case that the errors only affect a subset of the observations. We can model these errors through a multi-population model. Suppose that the bulk of the observations is well-explained by a low-rank model while the remainder come from another population or are corrupted by measurement noise. A prudent approach to robust principal component analysis would first separate the corrupted data from the uncorrupted data before attempting to recover a low-rank model. When the corrupted rows are unknown, this task may seem daunting.

To accomplish this task, we propose a semidefinite program that decomposes the input \mathbf{X} into two matrices:

$$\begin{aligned} & \underset{(\mathbf{P}, \mathbf{C})}{\text{minimize}} && \|\mathbf{P}\|_{2 \rightarrow 2}^* + \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^* \\ & \text{subject to} && \mathbf{P} + \mathbf{C} = \mathbf{X}. \end{aligned} \quad (3.1)$$

The norm $\|\mathbf{P}\|_{2 \rightarrow 2}^*$ is the sum of the singular values of \mathbf{P} and is known to promote low-rank solutions [15], while $\|\mathbf{C}\|_{2 \rightarrow \infty}^*$ is the sum of the ℓ_2 norms of the rows of \mathbf{C} and promotes group sparsity [37].

We call the optimal matrix pair $(\mathbf{P}_\star, \mathbf{C}_\star)$ for the problem (3.1) the *low-leverage decomposition* (LLD) of \mathbf{X} ; we can interpret \mathbf{C}_\star as an identified corruption and \mathbf{P}_\star as a surrogate for the uncorrupted observations. We define our robust components as the right singular vectors of the surrogate matrix \mathbf{P}_\star . The detailed procedure appears in Algorithm 2. We show in Section 3.1 that our recovered data matrix \mathbf{P}_\star has the additional property of being a low-leverage set of observations.

The LLD formulation is related to recent proposals [6, 7], and we discuss this point more in Section 4.2.

As we were preparing this manuscript, we became aware of the independent work [46, 47] which also considers (3.1) for the robust PCA problem. This work shows that, under certain hypotheses, the recovered low-rank data \mathbf{P}_\star has the same row-space as the true data and the corrupted rows are correctly identified.

3.1. Low-Leverage by Duality. In this section, we demonstrate that (3.1) extracts a low-leverage model for the data. This result follows from duality arguments that characterize the optimum of the convex program.

Lemma 3.1 (First-order optimality conditions for (3.1)). *A feasible pair (\mathbf{P}, \mathbf{C}) is optimal for (3.1) if and only if there exists a matrix \mathbf{Q} such that*

$$\langle \mathbf{Q}, \mathbf{P} \rangle = \|\mathbf{P}\|_{2 \rightarrow 2}^*, \quad \|\mathbf{Q}\|_{2 \rightarrow 2} \leq 1 \quad (3.3a)$$

$$-\langle \mathbf{Q}, \mathbf{C} \rangle = \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^*, \quad \|\mathbf{Q}\|_{2 \rightarrow \infty} \leq \gamma, \quad (3.3b)$$

Proof. It follows from standard subdifferential conditions that a feasible point (\mathbf{P}, \mathbf{C}) minimizes the functional in (3.1) if and only if zero is in the subgradient of $f(\mathbf{P}) = \|\mathbf{P}\|_{2 \rightarrow 2}^* + \gamma \|\mathbf{X} - \mathbf{P}\|_{2 \rightarrow \infty}^*$. By the additivity of subgradients [38, Thm. 23.8], this condition holds if and only if there exists a matrix \mathbf{Q} such that the subgradient conditions $\mathbf{Q} \in \partial \|\mathbf{P}\|_{2 \rightarrow 2}^*$ and $-\mathbf{Q} \in \partial \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^*$ are in force.

We show that these subgradient conditions are equivalent to (3.3). By definition of the subdifferential, $\mathbf{Q} \in \partial \|\mathbf{P}\|_{2 \rightarrow 2}^*$ if and only if for every perturbation Δ the subgradient inequality

$$\langle \mathbf{Q}, \Delta \rangle \leq \|\mathbf{P} + \Delta\|_{2 \rightarrow 2}^* - \|\mathbf{P}\|_{2 \rightarrow 2}^* \quad (3.4)$$

holds. Suppose first that (3.3a) holds. Then, for all Δ , we have

$$\langle \mathbf{Q}, \Delta \rangle = \langle \mathbf{Q}, \mathbf{P} + \Delta \rangle - \|\mathbf{P}\|_{2 \rightarrow 2}^* \leq \|\mathbf{Q}\|_{2 \rightarrow 2} \|\mathbf{P} + \Delta\|_{2 \rightarrow 2}^* - \|\mathbf{P}\|_{2 \rightarrow 2}^*,$$

where the inequality follows by the definition of dual norms. Since $\|\mathbf{Q}\| \leq 1$ by assumption, the subgradient inequality (3.4) must hold.

It remains to show that the subgradient inequality (3.4) implies (3.3a). Taking $\Delta = \mathbf{P}$ in (3.4) gives $\langle \mathbf{Q}, \mathbf{P} \rangle \leq \|\mathbf{P}\|_{2 \rightarrow 2}^*$, while $\Delta = -\mathbf{P}$ gives the reverse inequality $\langle \mathbf{Q}, \mathbf{P} \rangle \geq \|\mathbf{P}\|_{2 \rightarrow 2}^*$. Therefore the subgradient inequality (3.4) implies $\langle \mathbf{Q}, \mathbf{P} \rangle = \|\mathbf{P}\|_{2 \rightarrow 2}^*$.

On the other hand, suppose that $\Delta \neq \mathbf{0}$ satisfies $\langle \mathbf{Q}, \Delta \rangle = \|\mathbf{Q}\|_{2 \rightarrow 2} \|\Delta\|_{2 \rightarrow 2}^*$; such a matrix Δ must always exist in finite dimensions since suprema are attained in the trace definition of norms. Then the subgradient inequality (3.4) implies

$$\|\mathbf{Q}\|_{2 \rightarrow 2} \|\Delta\|_{2 \rightarrow 2}^* \leq \|\mathbf{P} + \Delta\|_{2 \rightarrow 2}^* - \|\mathbf{P}\|_{2 \rightarrow 2}^* \leq \|\Delta\|$$

where the second inequality follows by the triangle inequality. Since $\Delta \neq \mathbf{0}$, we have shown that the subgradient inequality implies $\|\mathbf{Q}\|_{2 \rightarrow 2} \leq 1$. Hence $\mathbf{Q} \in \partial \|\mathbf{P}\|_{2 \rightarrow 2}^*$ is equivalent to (3.3a). The equivalence between $-\mathbf{Q} \in \partial \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^*$ and relation (3.3b) follows analogously. \square

Before continuing, we introduce another fact concerning the subgradient of unitarily invariant norms. Let $\mathbf{P} = \mathbf{U}\Sigma\mathbf{V}^*$ be the compact SVD of \mathbf{P} . It follows from [44] that (3.3a) implies $\mathbf{Q} = \mathbf{U}\mathbf{V}^* + \mathbf{W}$, where, in particular, $\mathbf{U}\mathbf{V}^*\mathbf{W} = \mathbf{0}$.

3.1.1. Leverage scores. The *leverage score* of the observation \mathbf{x}_i corresponding to the i th row of \mathbf{X} is given by the number $[\mathbf{H}]_{ii}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}^*\mathbf{X})^\dagger\mathbf{X}^*$ is the orthoprojector onto the column space of \mathbf{X} . We refer to \mathbf{H} as the *hat matrix* in accord with common statistical practice. A large leverage score tends to indicate that the corresponding observation lies outside of the bulk of the data, although it does not necessarily indicate that the point is influential in linear regression. We refer to [32, Ch. 6] for further discussion of leverage scores.

The following theorem shows that the leverage scores of our decomposition are bounded above by γ^2 , justifying the terminology low-leverage decomposition for the solution of the program (3.1).

Theorem 3.2. *Suppose $(\mathbf{P}_*, \mathbf{C}_*)$ is an optimal point of the program (3.1). Then the diagonal elements of the hat matrix $\mathbf{H} = \mathbf{P}_*(\mathbf{P}_*^*\mathbf{P}_*)^\dagger\mathbf{P}_*^*$ are bounded above by γ^2 .*

Proof. From the characterization of the subgradient of unitarily invariant norms [44] discussed above, we know that $\mathbf{Q} = \mathbf{U}\mathbf{V}^* + \mathbf{W}$ with $\mathbf{U}\mathbf{V}^*\mathbf{W} = \mathbf{0}$. Thus,

$$\mathbf{Q}\mathbf{Q}^* = \mathbf{U}\mathbf{U}^* + \mathbf{W}\mathbf{W}^* \succcurlyeq \mathbf{U}\mathbf{U}^* = \mathbf{H},$$

where the last equality can be easily checked using the definition of \mathbf{H} and the SVD of \mathbf{P}_* . Since the diagonal entries of a positive-semidefinite matrix are nonnegative, this relation implies $[\mathbf{H}]_{ii} \leq [\mathbf{Q}\mathbf{Q}^*]_{ii}$. Recall that the $\ell_2 \rightarrow \ell_\infty$ operator norm is the maximum ℓ_2 row norm of the matrix. Thus relation (3.3b) of Lemma 3.1 implies that $[\mathbf{Q}\mathbf{Q}^*]_{ii} \leq \gamma^2$, which completes the proof. \square

We can view our proposal as a method of decomposing a data matrix \mathbf{X} into a component with a (user-specified!) upper bound on the leverage plus an error term. Moreover, this result gives a statistical interpretation to the regularization parameter γ in (3.1).

We note that while our program guarantees a low-leverage decomposition, an assumption of suitably small leverage is a technical hypothesis in other works, e.g., [6, eq. (1.2)].

The reader should be warned that this method does not necessarily produce a low-leverage solution if we use our program to identify outlying data and then “prune” the rows. That is, suppose $(\mathbf{P}_\star, \mathbf{C}_\star)$ is an optimal point of (3.1) and $\mathbf{c}_i = \mathbf{0}$ for row indices $i \in I$. Then the corresponding matrix $\mathbf{X}_I = \mathbf{P}_I$ *does not* necessarily have leverage scores bounded above by γ^2 .

3.2. The Choice of γ . In this section, we study how the value of the regularization parameter γ affects the properties of the decomposition.

We begin by showing that, when $\gamma \geq 1$, the degenerate solution $(\mathbf{P}_\star, \mathbf{C}_\star) = (\mathbf{X}, \mathbf{0})$ minimizes (3.1). This claim follows by explicit construction. Let $\mathbf{U}\Sigma\mathbf{V}^*$ be the compact SVD of \mathbf{X} , and define $\mathbf{Q} = \mathbf{U}\mathbf{V}^*$. Clearly $\langle \mathbf{Q}, \mathbf{X} \rangle = \|\mathbf{X}\|_{2 \rightarrow 2}^*$, so \mathbf{Q} satisfies (3.3a) with $\mathbf{P}_\star = \mathbf{X}$. By construction, the maximum singular value of \mathbf{Q} is bounded above by one. Equivalently, $\mathbf{Q}\mathbf{Q}^* \preceq \mathbf{I}$. This inequality implies $[\mathbf{Q}\mathbf{Q}^*]_{ii} \leq 1$. Since the diagonal entries of $\mathbf{Q}\mathbf{Q}^*$ are the squared row norms of \mathbf{Q} , we have shown that $\|\mathbf{Q}\|_{2 \rightarrow \infty} \leq 1 \leq \gamma$. This bound demonstrates that \mathbf{Q} satisfies (3.3b) with $\mathbf{C}_\star = \mathbf{0}$, which certifies optimality of this degenerate solution by Lemma 3.1.

We now show that the regularization parameter γ gives an upper bound on the rank of the optimal \mathbf{P}_\star . It is easy to show using the SVD of \mathbf{P}_\star that the trace of the hat matrix \mathbf{H} defined above is equal the rank of \mathbf{P}_\star . Since $[\mathbf{H}]_{ii} \leq \gamma^2$ by Theorem 3.2, we must have

$$\text{rank}(\mathbf{P}_\star) = \text{trace}(\mathbf{H}) \leq n\gamma^2. \quad (3.5)$$

The rank is a positive integer, so $\gamma < 1/\sqrt{n}$ implies that the optimal \mathbf{P}_\star is trivial. Moreover, in order to get T meaningful components in Step 2 of Algorithm 2, we require $\text{rank}(\mathbf{P}_\star) \geq T$. Thus, we can limit ourselves to situations where $\gamma \in [\sqrt{T/n}, 1]$.

Inequality (3.5) has implications for the numerical solution of (3.1). As we discuss in Section 3.3, the bulk of the computation comes from computing an SVD at each iteration. When the solution of the optimization problem has low rank, the iterates also tend to have low rank. This allows us to save significant computational effort by computing partial singular decompositions at each step. A judicious choice of γ can increase the performance of our algorithm immensely. We find that taking $n\gamma^2 \approx T^2$ is a useful heuristic for achieving a rank- T optimal solution, so long as $n \gg T^2$.

On the other hand, typical statistical data does not show true low-rank behavior even when there are no outliers. Therefore, forcing the optimal decomposition to be low rank typically results in a dense corruption \mathbf{C}_\star . This effect may be mitigated somewhat by another formulation we discuss briefly in Section 3.4. In practice we find that setting γ somewhat less than $\sqrt{p/n}$, say $\gamma = 0.8\sqrt{p/n}$, provides a very good low-rank model, but it does poorly in the context of outlier identification. We discuss specific parameter choices for our experiments in Section 5.

3.3. Computing the Low-Leverage Decomposition. Although general-purpose semidefinite programming software such as CVX [19, 18] can solve small instances of (3.1) efficiently, the interior-point methods they utilize may be unable to complete even a single iteration of a large-scale problem. This observation indicates that we need to use different methods for large-scale problems.

To solve (3.1), we recommend an alternating direction augmented Lagrangian algorithm analogous to the one used in [6]; see also [27]. The generic form of the method is known as the Augmented Lagrangian Method of Multipliers (ALMM). The augmented Lagrangian for (3.1) with dual variable \mathbf{Q} is given by

$$\mathcal{L}_\mu(\mathbf{P}, \mathbf{C}, \mathbf{Q}) = \|\mathbf{P}\|_{2 \rightarrow 2}^* + \gamma \|\mathbf{C}\|_{2 \rightarrow \infty}^* + \langle \mathbf{X} - \mathbf{P} - \mathbf{C}, \mathbf{Q} \rangle + \frac{\mu}{2} \|\mathbf{X} - \mathbf{P} - \mathbf{C}\|_{\text{F}}^2. \quad (3.6)$$

For an initial starting point \mathbf{P}^0 , we alternately solve $\mathbf{P}^{k+1} = \arg \min_{\mathbf{P}} \mathcal{L}_\mu(\mathbf{P}, \mathbf{C}^k, \mathbf{Q}^k)$ and $\mathbf{C}^{k+1} = \arg \min_{\mathbf{C}} \mathcal{L}_\mu(\mathbf{P}^{k+1}, \mathbf{C}, \mathbf{Q}^k)$. We then update the multiplier by the feasibility gap $\mathbf{Q}^{k+1} = \mathbf{Q}^k + \mu(\mathbf{X} - \mathbf{P}^{k+1} - \mathbf{C}^{k+1})$.

The minimizations above have an explicit form in terms of shrinkage operations [8]

$$\mathbf{C}^{k+1} = \text{RowShrink}\left(\mathbf{X} - \mathbf{P}^k + \frac{1}{\mu}\mathbf{Q}^k, \mu\gamma\right) \quad (3.7a)$$

$$\mathbf{P}^{k+1} = \text{SpecShrink}\left(\mathbf{X} - \mathbf{C}^{k+1} + \frac{1}{\mu}\mathbf{Q}^k, \mu\right), \quad (3.7b)$$

where $\text{RowShrink}(\mathbf{A}, \nu)$ soft-thresholds each row \mathbf{a}_i of \mathbf{A} :

$$\text{RowShrink}(:, \nu) : \mathbf{A} \mapsto \text{diag}([1 - \nu/\|\mathbf{a}_i\|_2]_+) \cdot \mathbf{A},$$

where $[x]_+ = \max\{x, 0\}$. Similarly $\text{SpecShrink}(\mathbf{A}, \nu)$ soft-thresholds the singular values of \mathbf{A}

$$\text{SpecShrink}(:, \nu) : \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \mapsto \mathbf{U}[\mathbf{\Sigma} - \nu\mathbf{I}]_+ \mathbf{V}^*, \quad (3.8)$$

where the operator $[\cdot]_+$ is applied element-wise. We initialize the algorithm with $\mathbf{P}^0 = \mathbf{0}$ and set the parameter $\mu = np/\|\mathbf{X}\|_{2 \rightarrow \infty}^*$. We stop the algorithm when the iterates are nearly feasible, that is, $\|\mathbf{X} - \mathbf{P}^k - \mathbf{C}^k\| < 10^{-7}\|\mathbf{X}\|_{\text{F}}$.

The main computational difficulty when running this algorithm involves computing the spectral shrinkage operator. When the iterates \mathbf{P}^k are low rank, we can save significant computational effort by performing only partial singular value decompositions [27]. We can leverage our analysis in Section 3.2 to ensure that the optimal \mathbf{P}_* is low rank. Since the algorithmic iterates tend to be low-rank in this case, we can significantly improve the performance of our algorithm by choosing γ to limit the rank of the optimal solution. In practice, we have found that one should set the quantity $n\gamma^2$ somewhat larger than the desired rank of the solution, e.g., $n\gamma^2 \approx T^2$ when we desire a rank- T solution.

3.4. Extensions for a Noisy Model. We note that there is an obvious extension of the LLD when one wants to account for an additional of noise in the model. Suppose that in addition to gross corruptions of certain observations, we would also like to model small corruptions or noise that may be spread throughout the data.

Instead of enforcing the equality $\mathbf{X} = \mathbf{P} + \mathbf{C}$, we allow for some additional slack of the form $\|\mathbf{X} - \mathbf{P} - \mathbf{C}\|_{\text{F}} \leq \eta$, where η is an estimate for the noise level. That is, we solve the problem

$$\begin{aligned} & \text{minimize} && \|\mathbf{P}\|_{2 \rightarrow 2}^* + \gamma\|\mathbf{C}\|_{2 \rightarrow \infty}^* \\ & \text{subject to} && \|\mathbf{X} - \mathbf{P} - \mathbf{C}\|_{\text{F}} \leq \eta \end{aligned} \quad (3.9)$$

When $\eta = 0$, this is equivalent to our proposal (3.1) for the gross corruption model. Other loss functions are also possible. Note that the Frobenius norm remains invariant under a rotation on the right, which is a feature of (3.1) that we would like to preserve.

This formulation is also studied in the independent work [46, 47]. It is shown there that under some technical conditions, the decomposition from (3.9) results in a decomposition where \mathbf{P}_* is close to a matrix with the same row-space as the true observations, and the matrix \mathbf{C}_* is close to a matrix that correctly identifies the column support of the corruption.

4. PREVIOUS WORK

This section describes previous work on robust formulations of principal component analysis. Convex approaches to robust PCA are unusual, and, as a consequence, many other attempts at robust PCA lack rigorous algorithms. Often, proposals are put forward with a mathematical formulation and only a heuristic algorithm—or an algorithm without a clear mathematical formulation.

In Sections 4.1 and 4.2, we describe the two methods in the literature most closely related to our proposals. We then describe in detail an approach for robust PCA recommended by Maronna [29] with which we provide comparisons in Section 5. We conclude with a short overview of other robust PCA proposals that have appeared in the literature.

4.1. Antecedents for MDR: Projection Pursuit PCA. Our MDR proposal is a particular instance of an approach that has come to be known as *projection-pursuit PCA* (PP-PCA), as we discuss in Section 2.2. The theoretical properties of PP-PCA are well understood; see for instance [12] and [11].

All of the algorithms we have found in the literature for computing PP-PCA are meant to operate with an arbitrary scale. In view of the fact that the PP-PCA problem is NP-hard, it is unsurprising that the literature appears to contain no PP-PCA algorithms with proofs of correctness and tractability. Indeed, we have been unable to find other work that recognizes that the PP-PCA problem is intractable in a rigorous sense.

The original study of Li and Chen [26] uses a Monte Carlo approach that was found to be computationally expensive. In theory, even simple Monte Carlo methods (e.g., randomly sampling the unit sphere) can produce arbitrarily good solutions to problem (2.2) with an arbitrary (continuous) scale. Given the computational hardness of the problem, it is unlikely that Monte Carlo approaches can provide guarantees of computational efficiency.

Current algorithms for PP-PCA rely on heuristics. A popular and fast algorithm for generic projection-pursuit PCA is the finite direction method (FDM) of Croux and Ruiz-Gazen [11]. This technique replaces the search over the entire unit sphere $\|\mathbf{v}\|_2 = 1$ with a finite search over the directions that appear among the observations: $\mathbf{v} \in \{\mathbf{x}_1/\|\mathbf{x}_1\|_2, \dots, \mathbf{x}_n/\|\mathbf{x}_n\|_2\}$. The hope is that directions of large scale are likely to be well approximated by directions appearing in the data. This heuristic performs poorly when n and p are large because it takes an extremely large number of points to cover a high-dimensional sphere.

4.2. A convex approach. Recently, a method of Chandrasekaran et al. [7] has been adapted for robust PCA in [6]. This approach attempts to decompose the data matrix into a sum of a low-rank matrix and a sparse matrix via the semidefinite program

$$\begin{aligned} & \text{minimize} && \|\mathbf{L}\|_{2 \rightarrow 2}^* + \lambda \|\mathbf{S}\|_{1 \rightarrow \infty}^* \\ & \text{subject to} && \mathbf{L} + \mathbf{S} = \mathbf{X}. \end{aligned} \tag{4.1}$$

The nuclear norm $\|\cdot\|_{2 \rightarrow 2}^*$ promotes low rank and the matrix ℓ_1 norm $\|\cdot\|_{1 \rightarrow \infty}^*$ promotes sparsity. We refer to this method as N + L1. The works [6, 7] provide conditions under which N + L1 succeeds in *exactly* recovering a low-rank and sparse component.

This convex approach is principled in the sense that the mathematical formulation is also algorithmically tractable. On the other hand, it lacks an invariance to a change in the observation basis possessed by all other methods we discuss, including standard PCA. That is, applying a rotation $\mathbf{U}^* \mathbf{U} = \mathbf{I}$ to the data $\widehat{\mathbf{X}} = \mathbf{X} \mathbf{U}$ does not result in a similar rotation of the decomposition due to the fact that the norm $\|\cdot\|_{1 \rightarrow \infty}^*$ is not invariant under this transformation.

One may argue that this invariance is inconsequential: in real data, the particular choice of coordinates has a meaning and outliers may occur coordinate-wise. This argument is defensible in domain specific examples, such as image data that contain specularities [6]. Nevertheless, PCA is intended to locate a coordinate basis that explains data more effectively than the standard basis [23]. If this is the analytical goal, basis invariance is indeed a requisite property. See Section 5.1.2 for an experiment where this lack of orthogonal invariance in N + L1 appears to produce unnerving results.

4.3. Spherical PCA. Another approach, known as spherical principal components (sphPCA) [28], rescales the observations to unit (Euclidean) norm and applies standard PCA to this modified data. To implement the sphPCA method, we first compute a normalized matrix $\widehat{\mathbf{X}}$. Each row of $\widehat{\mathbf{X}}$ is the normalized version of the corresponding row of the centered data matrix \mathbf{X} , that is $\widehat{\mathbf{x}}_i = \mathbf{x}_i / \|\mathbf{x}_i\|_2$. Using the row-normalization operator from (2.11), we can express the normalized matrix as $\widehat{\mathbf{X}} = \mathcal{N}(\mathbf{X})$.

The robust components are then defined as the standard principal components of the rescaled matrix $\widehat{\mathbf{X}}$. Since all of the observations from the normalized matrix $\widehat{\mathbf{X}}$ have norm one, there are no large magnitude observations that exert an undue influence on the principal components.

A study by Maronna [29] shows that sphPCA enjoys good practical performance. The ease of implementation and relatively good behavior of sphPCA leads Maronna to suggest it as the

default choice for robust principal component analysis. As a result, we use sphPCA as a baseline comparison for the performance of our robust methods in Section 5.

4.4. Other proposals. Some of the earliest methods for robust PCA compute approximations of correlation or covariance matrices using robust methods. Gnanadesikan and Kettenring propose direct robust estimation of the covariance matrices through robust estimation of the individual entries [17]. This may lead to counterintuitive results such as non-positive covariance matrices. An alternative approach explicitly enforces positive matrices as minimizers of a functional such as an M -estimator [14]; see also the more recent study [10].

A representative example of robust PCA from the machine learning community is the work of De La Torre and Black [13]. They define the robust components as the minimum of a highly non-convex energy function and attempt to minimize this energy function using an iteratively reweighted least-squares algorithm coupled with an annealing step. No theoretical guarantees of correctness for the algorithm are provided.

Another recent approach appears in the paper [45] of Xu et al. This algorithm randomly removes observations that appear to have high influence in the current estimate of the principal components. The principal component estimate is computed from the trimmed data. Xu et al. are able to establish strong theoretical properties of their algorithm, including a high breakdown point in the high-dimensional scaling regime where $n \rightarrow \infty$ and $n/p \rightarrow c > 0$.

5. NUMERICAL EXPERIMENTS

This section provides some numerical examples comparing our proposals with standard PCA and other robust PCA methods in the literature. In Section 5.1, we look at the projection of two data sets on the top robust component. Section 5.2 repeats a multiple-component experiment of Maronna [30] with additional robust methods. Section 5.3 contains a larger experiment, where we calculate the first two components of a dense matrix with more than twenty million entries.

All of these experiments and algorithms are implemented with MATLAB. Following the principle of reproducible research [3], we provide code that reproduces the exact experiments in this work [31].

5.1. Projection onto the top component. In this section, we study the robust component methods applied to two data sets. The first set is a selection of environmental factors that may affect the concentration of nitrogen dioxide around Oslo, Norway. The second example is constructed from standard iris data. In each case, we examine the spread of the data in the direction of the top robust component.

5.1.1. Experimental setup. For these experiments, we center the data by removing the Euclidean median from each observation. The Euclidean median $\hat{\mu}$ is a robust estimate of the center of the data, and is defined as

$$\hat{\mu} = \arg \min_{\mu} \sum_{i=1}^n \|x_i - \mu\|_2. \quad (5.1)$$

Maronna [30, Ch. 9] gives a method to solve this convex problem for $\hat{\mu}$.

We project the data onto the top component for each method and compare the performance of the methods by the *interquartile range* (IQR), that is, the distance between the 25th and 75th percentile of the projected data.

We apply extract the dominant component from each data set using our methods (MDR and LLD), other robust methods (sphPCA and N + L1), and standard PCA. For MDR, we use $K = 94$ rounding trials as discussed in Section 2.4. We set the LLD weight parameter $\gamma = 0.8\sqrt{p/n}$. As recommended in [6], we set the N + L1 parameter $\lambda = 1/\sqrt{n}$ for the first experiment. With the iris data in Section 5.1.3, we find that $\lambda = 1/\sqrt{n}$ gives a trivial result: no outliers were identified by N + L1. Instead, we use the more favorable choice $\lambda = 0.3/\sqrt{n}$.

5.1.2. Norwegian nitrogen dioxide data. Our data for this experiment consists of 500 observations of eight environmental factors around Oslo, Norway, available on the Statlib archive [1]. The variables include the log-concentration of nitrogen dioxide (NO₂) particles, the number of cars per hour, and the wind speed, as well as several additional factors useful for predicting the concentration of NO₂ particles.

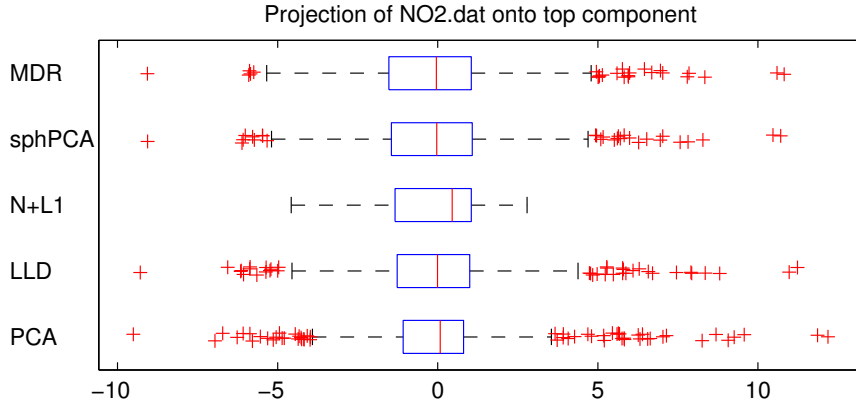


FIGURE 1. *Projection of the Oslo NO₂ data set onto top components.* The box surrounds the middle 50% of the data. The vertical line in the box is the median of the data. Each whisker extends either 1.5 times the length of the IQR or to the extreme value of the data, and the red crosses beyond the whiskers are the outlying points. The plots are ordered by decreasing IQR.

We calculate the top component of the data using each method. In Figure 1 we plot the projection of the data onto the direction of these components using a standard box-and-whisker plot. The whiskers extend either 1.5 times the IQR beyond the edge of the box or to the extreme data point. We consider points that lie beyond the whiskers outliers. We give the percentage of outliers and several order statistics of the data in Table 2.

Every robust method results in a larger IQR than PCA. The MDR component finds the largest IQR, and the LLD method finds the smallest IQR among the robust methods. Except for N + L1, every method identifies a direction with a relatively large number of outliers, which indicates that the data has heavy tails.

The N + L1 method is unique because it does not identify a direction of large spread *outside* of the middle 50% of the data. We have observed that a random change of the observation basis causes the N + L1 component to perform similarly to the LLD component. By orthogonal equivariance, the results for methods other than N + L1 are unchanged by a change in the observation basis. This indicates that the behavior of the results given by the N + L1 component is due to the lack of orthogonal equivariance.

We note that the approximation ratio for the top MDR component is near optimal at 0.978.

TABLE 2. *Statistics for the projected NO₂ data.* The last column lists the percentage of points lying outside the whiskers in Figure 1.

Method	IQR	min	25th	75th	max	out
MDR	2.57	-9.07	-1.53	1.05	10.82	5.00%
sphPCA	2.53	-9.06	-1.45	1.08	10.71	5.60%
N+L1	2.38	-4.58	-1.34	1.05	2.79	0.00%
LLD	2.27	-9.29	-1.27	1.00	11.24	7.40%
PCA	1.89	-9.51	-1.08	0.81	12.18	11.00%

5.1.3. *Iris data.* We use Fisher's iris data [16] in this experiment. The data contains 60 observations from three different species of iris: *Iris setosa*, *Iris virginica*, and *Iris versicolor*. Each observation consists of four measurements, namely sepal length, sepal width, petal length, and petal width.

Fifty of the observations come from the *setosa* flowers. We corrupt these observations with 5 measurements of *Iris virginica* and five measurements of *Iris versicolor*. We hope that robust principal components identify a direction of large spread in the *setosa* bulk of the data.

As a baseline comparison, we also calculate the dominant principal component of the *setosa* population without the outlying flowers.

As in Section 5.1.2, we project the data onto the direction of the dominant components. These points are plotted in Figure 2; we distinguish the bulk *setosa* points from the *versicolor* and *virginica* observations. We compute an approximate density of the *setosa* observations by convolving the projected data with a unit volume Gaussian kernel of width $\sigma = 0.2$. Table 3 gives some order statistics of the projections.

TABLE 3. *Order statistics for the projection of the setosa data onto the top components.* The last column lists the number of *setosa* points further than 1.5 times IQR left of the 25th percentile or the right of the 75th percentile.

Method	IQR	min	25th	75th	max	out
LLD	0.70	-1.21	-0.41	0.29	1.14	0.00%
<i>Setosa</i> PCA	0.70	-1.22	-0.41	0.29	1.14	0.00%
sphPCA	0.69	-1.19	-0.41	0.28	1.13	0.00%
N+L1	0.66	-1.16	-0.40	0.26	1.07	0.00%
MDR	0.37	-0.79	-0.24	0.13	0.53	0.00%
PCA	0.19	-0.60	-0.15	0.04	0.37	6.00%

The dominant component of LLD, sphPCA, and N + L1 each achieves an IQR at least 3 times that of PCA. These components do not clearly distinguish among the three populations, indicating that these methods are insensitive to the effect of the outliers. LLD and sphPCA appear the most effective in this situation; indeed, it appears that LLD and sphPCA perform as well as *setosa*-only PCA.

Although MDR results in the most modest IQR in the *setosa* among the robust methods, the IQR associated with the MDR component is 1.95 times the IQR of the *setosa* family along the dominant PCA component. Unlike the other robust methods, the MDR component discriminates among the three distinct populations. While it is clear that MDR *does not* reject the influence of the outliers, MDR balances the influence of outliers and the bulk of the data better than PCA. In this experiment the optimality ratio for MDR is 0.9975, certifying that the MDR component is essentially the direction of maximum mean deviation in the data.

5.2. Regression Surface for Bus Data. In this experiment, we construct a regression surface using multiple components. A point is well described by a surface if its Euclidean distance from the surface is small. The dominant T classical principal components span a T -dimensional regression surface such that the sum of the squared distances of the observations to the plane is minimized. We would hope that robust components describe the bulk of the points better than standard components when outliers contaminate the data. We illustrate this behavior with an experiment of Maronna et al. [30, p. 214], which we augment with additional robust methods.

5.2.1. Experimental setup. Our data consists of $p = 18$ geometric features collected from $n = 218$ bus silhouettes [40] that we arrange into an $n \times p$ matrix \mathbf{X} . Following Maronna et al., we remove the 9th variable from the data and divide the columns of \mathbf{X} by their median absolute deviation (MADN), a robust measure of scale defined as

$$\text{MADN}(\mathbf{x}) = \text{median}(|\mathbf{x} - \text{median}(\mathbf{x})|).$$

We then center the observations by their Euclidean median. We compute the top three components using PCA, MDR, LLD, sphPCA, and N + L1. We take the LLD parameter $\gamma = 0.8\sqrt{n/m}$, the N + L1 parameter $\lambda = \sqrt{1/m}$, and the rounding count of MDR $K = 94$.

For each method, we determine the Euclidean distance from each observation to the orthogonal regression plane spanned by the dominant three components. In Figure 3, we plot the ordered distances to the robust hyperplanes against the ordered distances to the PCA hyperplane.

Since the PCA regression surface minimizes the sum of squared distances to the observations, not all of the observations can lie below the 1:1 line. However, a large number of points below

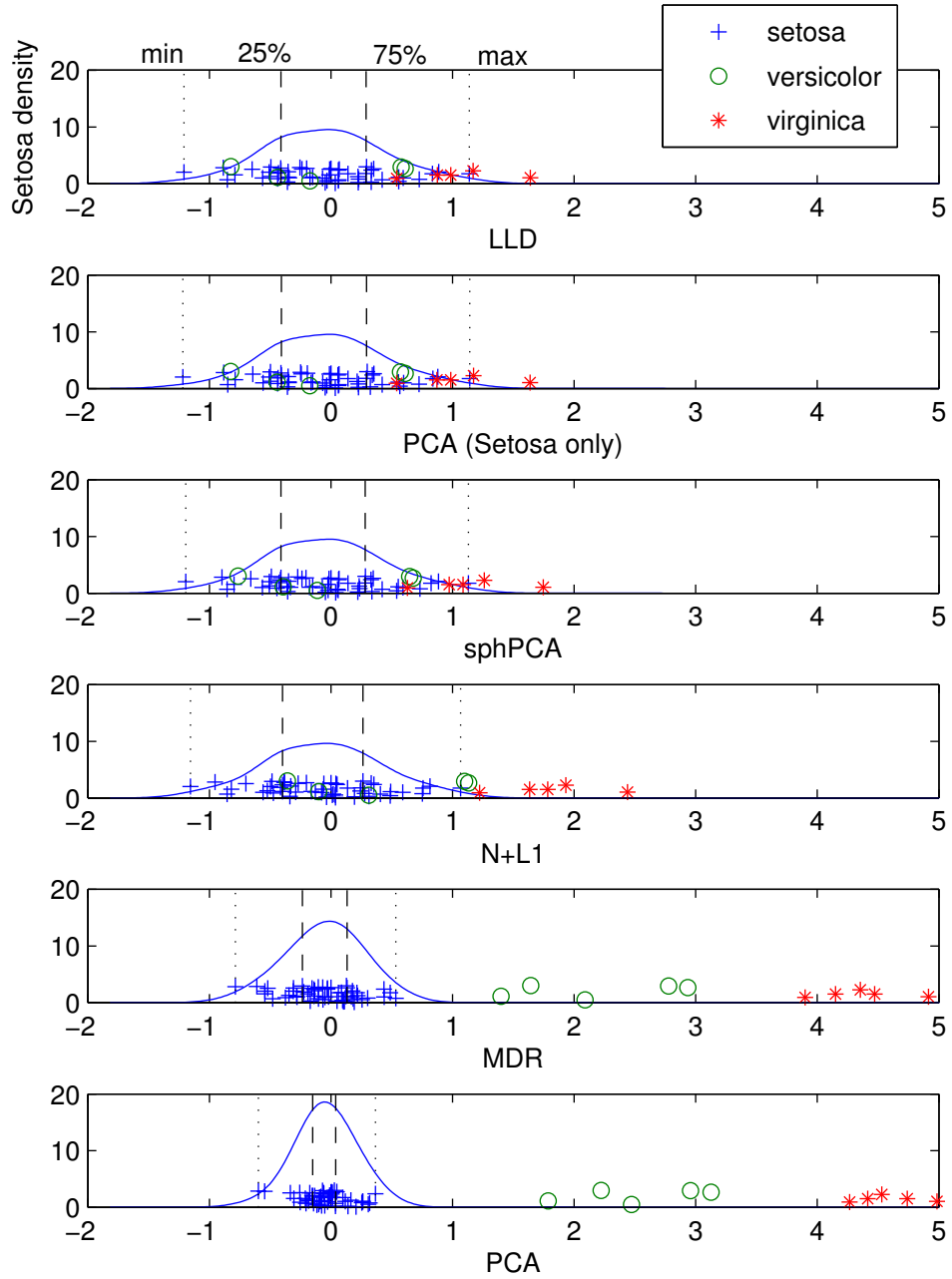


FIGURE 2. The projections of the iris data onto the top components. The points are randomly jittered above the zero line for readability. The blue curve represents the approximate local point density of *setosa*. Note that LLD and sphPCA essentially provide the same projection as PCA *without outliers*. We sort the plots by decreasing IQR.

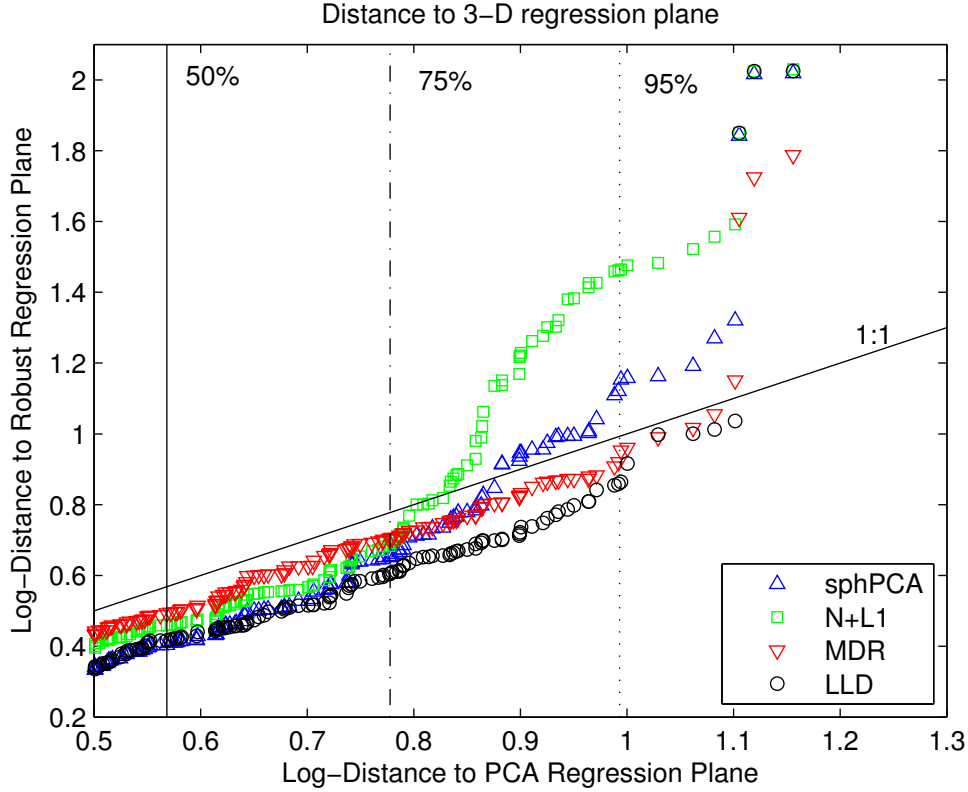


FIGURE 3. The distance of points to robust regression surfaces as a function of the distance of points to the standard PCA regression surface. The regression surface is determined by the top three components from each method. Points to the left of the median follow the same generic pattern as points in the 3rd quantile, and are therefore omitted. Three extreme points to the right are also omitted.

the 1:1 line indicates that a robust regression surface explains the bulk of the data better than the classical surface.

5.2.2. *Discussion.* Figure 3 focuses on the third and fourth quantiles of the data; the first and second quantiles roughly follow the pattern apparent in the third quantile. For clarity, we omit the three most outlying points that would appear in the upper right corner of the figure. Each robust method results in a regression surfaces that explains the data better than PCA for more than 75% of the points. In the third quantile, both $N + L1$ and sphPCA lose their explanatory advantage over PCA. It is not until the after 95% of the data that MDR and LLD provide worse explanations than PCA. LLD is the dominating method through the latter part of the data.

MDR explains the bulk of the data less effectively than the other robust methods, yet the final outlying observations are explained better by MDR than the other methods. This indicates that MDR is more sensitive to outlying points than the other robust methods, but is less sensitive to outliers than standard PCA. The optimality ratios for the first three MDR components are, respectively, 0.99999, 0.99992, and 0.97253, implying that MDR essentially succeeds in PP-PCA with the MD scale for this data.

Finally, we note that changing the $N + L1$ parameter to $\lambda = 2\sqrt{1/n}$ results in performance similar to LLD.

5.3. **Movielens.** We finish this section with a larger example: the million-rating movielens data [21]. The data consist of 6040 users rating and 3952 movies, though several movies are

TABLE 4. *Most important movies as given by the first standard principal component.* The decimal numbers represent the weight each component puts on a movie. The integer to the right of the weight is the rank of the movie under the given component.

Movie	PCA		MDR		SPH		LLD	
GoodFellas	0.0708	1	0.1016	1	0.1092	1	0.0885	1
Army of Darkness	0.0697	2	0.0914	3	0.0970	4	0.0832	2
A Little Princess	0.0664	3	0.0899	4	0.1028	2	0.0826	3
Pushing Hands	0.0657	4	0.0772	11	0.0827	10	0.0745	8
Stand by Me	0.0656	5	0.0853	5	0.0896	5	0.0765	5

TABLE 5. *Most important movies: second component.*

Movie	PCA		MDR		SPH		LLD	
Nikita	0.0982	1	0.1099	2	0.0959	13	0.1071	5
Citizen Kane	0.0945	2	0.1051	4	0.0935	15	0.1104	3
Fried Green Tomatoes	0.0917	3	0.0934	8	0.0727	35	0.0944	11
Unforgiven	0.0891	4	0.0923	10	0.0877	21	0.0982	9
Mommie Dearest	−0.0855	5	−0.1108	1	−0.1522	1	−0.1281	1

replicated. The set contains just over one million ratings. Each rating is between one and five stars, and each user in the data set rated at least 20 movies.

We arrange these responses into an $n = 6040$ by $p = 3952$ matrix \mathbf{X} whose rows correspond to the users and whose columns correspond to the movies. We set unrated movies to the user’s median rating, and center each user’s ratings by their personal median. As with our other experiments, we center the rows by the Euclidean median, which results in a dense matrix with nearly 24 million entries.

We then compute the top two components using PCA, MDR, LLD, and sphPCA. In order to speed up processing for LLD, we set $\gamma = \sqrt{100/p}$. As discussed in Section 3.3, this choice of γ limits the rank of the iterates $\mathbf{P}^{(k)}$ in the ALMM algorithm, which allows us to compute a partial SVD at each step. Our choice $\gamma = \sqrt{100/n}$ results in iterates whose rank is roughly 10; the rank of the optimal point \mathbf{P}_\star is nine.

Each component \mathbf{v} represents a direction in movie coordinates. The magnitude entry $[\mathbf{v}]_i$ indicates how much \mathbf{v} points in the direction of movie i . We use these magnitude of the entries in the components to rank the movies. We call movies with large magnitudes “important,” and we call the corresponding entry of the component a movie’s “importance.”

5.3.1. *Discussion.* Table 4 displays the five most important movies identified by the first standard principal component, along with the importance and rank calculated assigned to these movies by the robust components. Each method agrees that the violent mobster movie *GoodFellas* is the most important film. Indeed, *GoodFellas*, *Army of Darkness*, *A Little Princess*, and *Stand by Me* are ranked in the top five movies by every method. However, PCA ranks *Pushing Hands* much higher than the robust methods.

In Table 4, each importance has positive sign. For each method, the first component assigns very few movies a negative importance for the first component. This fact comes about because the typical user rating is positive; that is, the sum $\sum_j [\mathbf{x}_i]_j$ is greater than zero for most users.

Table 5 displays the results for the second components. Each robust component views *Mommie Dearest* as the most important movie, while standard PCA relegates it to fifth place. Neither *Fried Green Tomatoes* nor *Unforgiven* are among the top five movies for the robust methods. With the second component, sphPCA takes the most dramatic shift away from PCA, with only *Mommie Dearest* making it into the top ten movies.

Of course, rankings are not the whole story. The signs are very consistent³ between methods. *Mommie Dearest* is negative for every method considered and *Fried Green Tomatoes* is positive. The sign consistency indicates that these components are measuring essentially the same thing.

The magnitude of the importance are also telling. PCA assigns the smallest weight to every movie, with the exception of the second component of sphPCA. This indicates that the robust methods are willing to assign more importance to discriminating movies.

ACKNOWLEDGMENTS

The first author would like to thank Alex Gittens, Richard Chen, and Stephen Becker for valuable discussions regarding this work.

APPENDIX A. PROOF OF THEOREM 2.2

This appendix contains the proof of Theorem 2.2 that we repeat below as Theorem A.4. We begin with some supporting results. The following result of Alon and Naor [2, Sec. 4.2] allows us to bound the expectation of $\|\mathbf{X}\mathbf{v}_\star\|_1$ below. The essence of this result goes back to a 1953 paper of Grothendieck [20]; see also the little Grothendieck theorem in [36, Sec. 5b].

Lemma A.1. *Let α_\star^2 be the value of the optimization problem (2.10) of Algorithm 1. Then $\alpha_\star^2 \geq \|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1}$. Moreover, let $\mathbf{y}^{(k)}$ be one of the vectors generated in Step 2. Then $\mathbb{E} \|\mathbf{X}^* \mathbf{y}^{(k)}\|_2^2 \geq \frac{2}{\pi} \alpha_\star^2$.*

The claim $\alpha_\star^2 \geq \|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1}$ also follows from our discussion of the SDP relaxation in Section 2.4.1. We also need the following proposition.

Proposition A.2. *For each matrix \mathbf{X} , the identity $\|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1} = \|\mathbf{X}\|_{2 \rightarrow 1}^2$ holds.*

Proof. We can express

$$\|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1} = \max_{\substack{\|\mathbf{w}\|_\infty=1 \\ \|\mathbf{y}\|_\infty=1}} \langle \mathbf{X}^* \mathbf{w}, \mathbf{X}^* \mathbf{y} \rangle.$$

By the conditions for equality in the Cauchy–Schwarz inequality, it follows that we can take $\mathbf{w} = \mathbf{y}$ above. Hence

$$\|\mathbf{X}\mathbf{X}^*\|_{\infty \rightarrow 1} = \|\mathbf{X}^*\|_{\infty \rightarrow 2}^2 = \|\mathbf{X}\|_{1 \rightarrow 2},$$

where the last equality is a standard fact concerning adjoint operators. \square

We use the following variant of the Paley–Zygmund integral inequality [35] to bound the probability that $\|\mathbf{X}\mathbf{v}_\star\|_1$ is less than its expectation.

Lemma A.3. *Suppose Z is a random variable such that $0 \leq Z \leq C$ for some $C > 0$. Then, for any scalar $\theta \in [0, 1]$, we have $\mathbb{P}(Z > \theta \mathbb{E}[Z]) \geq C^{-1}(1 - \theta) \mathbb{E}[Z]$.*

Proof. Split the integral $\mathbb{E}[Z]$ into two integrals, the first over the region $Z \leq \theta \mathbb{E}[Z]$ and the second over the region $Z > \theta \mathbb{E}[Z]$. Notice that the former integral is bounded above by $\theta \mathbb{E}[Z]$, while the latter integral is bounded above by $C \mathbb{P}(Z > \theta \mathbb{E}[Z])$. Simple algebraic manipulation then shows the claim. \square

We now restate and prove the main Theorem of Section 2.

Theorem A.4. *Suppose that \mathbf{X} is an $n \times p$ matrix, and let K be the number of rounding trials. Let $(\mathbf{v}_\star, \alpha_\star)$ be the output of Algorithm 1. Then $\alpha_\star \geq \|\mathbf{X}\|_{2 \rightarrow 1}$. Moreover, for $\theta \in [0, 1]$, the inequality*

$$\|\mathbf{X}\mathbf{v}_\star\|_1 > \theta \sqrt{\frac{2}{\pi}} \alpha_\star$$

holds except with probability $e^{-2K(1-\theta^2)/\pi}$.

³Since components are only defined up to a sign, we mean that the sign pattern in Tables 4 and 5 are equivalent modulo multiplication by -1 .

Proof. Let $\mathbf{y} \in \{\pm 1\}^n$ be a sign vector and define $\mathbf{v} = \mathbf{X}^* \mathbf{y} / \|\mathbf{X}^* \mathbf{y}\|_2$. Then

$$\|\mathbf{X} \mathbf{v}\|_1 = \|\mathbf{X}^* \mathbf{y}\|_2^{-1} \max_{\mathbf{w} \in \{\pm 1\}^n} \langle \mathbf{w}, \mathbf{X} \mathbf{X}^* \mathbf{y} \rangle \geq \|\mathbf{X}^* \mathbf{y}\|_2$$

where the inequality follows by taking the specific choice $\mathbf{w} = \mathbf{y}$. In particular, this relation implies that the vectors $\mathbf{v}^{(k)} = \mathbf{X}^* \mathbf{y}^{(k)} / \|\mathbf{X}^* \mathbf{y}^{(k)}\|_2$ generated in Step 2 of Algorithm 1 satisfy

$$\mathbb{E} \|\mathbf{X} \mathbf{v}^{(k)}\|_1^2 \geq \mathbb{E} \|\mathbf{X}^* \mathbf{y}^{(k)}\|_2^2 \geq \frac{2}{\pi} \alpha_\star^2, \quad (\text{A.1})$$

where the last inequality follows from the second claim in Lemma A.1.

Since $\|\mathbf{v}^{(k)}\|_2 = 1$, the quantity $\|\mathbf{X} \mathbf{v}^{(k)}\|_1^2$ is a positive random variable bounded above by $\|\mathbf{X}\|_{2 \rightarrow 1}^2$. Therefore, inequality (A.1) and Lemma A.3 imply that

$$\mathbb{P} \left(\|\mathbf{X} \mathbf{v}^{(k)}\|_1^2 > \theta^2 \cdot \frac{2\alpha_\star^2}{\pi} \right) \geq (1 - \theta^2) \cdot \frac{2}{\pi} \cdot \left(\frac{\alpha_\star}{\|\mathbf{X}\|_{2 \rightarrow 1}} \right)^2 \geq \frac{2}{\pi} \cdot (1 - \theta^2), \quad (\text{A.2})$$

where we have used the fact that $\alpha_\star \geq \|\mathbf{X}\|_{2 \rightarrow 1}$ by Proposition A.2 and the first claim of Lemma A.1.

In Step 3 of the algorithm we have chosen \mathbf{v}_\star to maximize $\|\mathbf{X} \mathbf{v}_\star\|_1^2$, so the inequality $\|\mathbf{X} \mathbf{v}_\star\|_1^2 \leq 2(1 - \theta^2)/\pi$ holds if and only if $\|\mathbf{X} \mathbf{v}^{(k)}\|_1 \leq 2(1 - \theta^2)/\pi$ for all k . Therefore, the independence of $\mathbf{v}^{(k)}$ for $k = 1, \dots, K$ implies

$$\mathbb{P} \left(\|\mathbf{X} \mathbf{v}_\star\|_1 \leq \theta \sqrt{\frac{2}{\pi}} \|\mathbf{X}\|_{2 \rightarrow 1} \right) \leq \left(1 - \frac{2}{\pi} \cdot (1 - \theta^2) \right)^K < e^{-2K(1 - \theta^2)/\pi},$$

which completes the claim. \square

REFERENCES

- [1] M. Aldrin. NO2.dat. <http://lib.stat.cmu.edu/datasets/NO2.dat>, 2004.
- [2] N. Alon and A. Naor. Approximating the Cut-Norm via Grothendieck's Inequality. *SIAM J. Comput.*, 35(4):787, 2006.
- [3] J. Buckheit and D. Donoho. Wavelab and reproducible research, 1995.
- [4] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003.
- [5] S. Burer and R. D. C. Monteiro. Local Minima and Convergence in Low-Rank Semidefinite Programming. *Math. Program.*, 103(3):427–444, Dec. 2004.
- [6] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *preprint*, Dec. 2009. arXiv:0912.3599.
- [7] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-Sparsity Incoherence for Matrix Decomposition. *preprint*, June 2009. arXiv:0906.2220.
- [8] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2006.
- [9] C. Croux, P. Filzmoser, and M. R. Oliveira. Algorithms for projection-pursuit robust principal component analysis. *Chemom. Intell. Lab. Syst.*, 87:218–225, 2007.
- [10] C. Croux and H. Haesbroeck. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87(3):603–618, Sept. 2000.
- [11] C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *J. Multivariate Anal.*, 95(1):206–226, 2005.
- [12] H. Cui. Asymptotic distributions of principal components based on robust dispersions. *Biometrika*, 90(4):953–966, Dec. 2003.
- [13] F. De La Torre and M. Black. A framework for robust subspace learning. *Int. J. Comput. Vision*, 54(1):117–142, 2003.
- [14] S. Devlin, R. Gnandesikan, and J. Kettenring. Robust estimation of dispersion matrices and principal components. *J. Am. Stat. Assoc.*, 76(374):354–362, 1981.
- [15] M. Fazel. *Matrix rank minimization with applications*. Dissertation, Stanford University, Stanford, CA, 2002.
- [16] R. A. Fischer. The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, 7:179–188, 1936.
- [17] R. Gnandesikan and J. R. Kettenring. Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, 28(1):81–124, 1972.
- [18] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, London, 2008. http://stanford.edu/~boyd/graph_dcp.html.
- [19] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. <http://cvxr.com/cvx>, Oct. 2010.

- [20] A. Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques (French). *Bol. Soc. Mat. So Paulo*, 8:1–79, 1953.
- [21] Grouplens Research. MovieLens Data Sets. http://www.grouplens.org/system/files/million-ml-data.tar__0.gz.
- [22] W. W. Hager and H. Zhang. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Software*, 32(1):137, 2006.
- [23] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.
- [24] P. J. Huber. *Robust statistics*. Wiley, Hoboken, New Jersey, first edition, 1981.
- [25] P. J. Huber and E. Ronchetti. *Robust statistics*. Wiley, Hoboken, New Jersey, second edition, 2009.
- [26] G. Li and Z. Chen. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. *J. Am. Stat. Assoc.*, 80(391):759–766, 1985.
- [27] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Math. Program.*, submitted, 2009. arXiv:1009.5055.
- [28] N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, K. L. Cohen, G. Boente, R. Fraiman, B. Brumback, C. Croux, J. Fan, A. Kneip, J. I. Marden, D. Peña, J. Prieto, J. O. Ramsay, M. J. Valderrama, and A. M. Aguilera. Robust principal component analysis for functional data. *Test*, 8(1):1–73, June 1999.
- [29] R. A. Maronna. Principal Components and Orthogonal Regression Based on Robust Scales. *Technometrics*, 47(3):264–273, Aug. 2005.
- [30] R. A. Maronna, D. R. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2006.
- [31] M. McCoy and J. A. Tropp. Online code, 2010. <http://www.acm.caltech.edu/~mccoy/>.
- [32] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2006.
- [33] A. Nemirovski. Sums of random symmetric matrices and quadratic optimization under orthogonality constraints. *Math. Program.*, 109:283–317, January 2007.
- [34] Y. E. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.*, 9(1):141–160, 1998.
- [35] R. E. A. C. Paley and A. Zygmund. A note on analytic functions in the unit circle. *Math. Proc. Cambridge Philos. Soc.*, 28(03):266–272, Oct. 1932.
- [36] G. Pisier. *Factorization of linear operators and geometry of Banach spaces*. Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI, 1986.
- [37] B. D. Rao and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *Proceedings of the 8th IEEE Digital Signal Processing Workshop*, 1998.
- [38] R. T. Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, 1970.
- [39] J. Rohn. Computing the $\|\cdot\|_{\infty \rightarrow 1}$ Norm is NP-hard. *Linear and Multilinear Algebra*, 47(3):195–204, 2000.
- [40] J. P. Siebert. Vehicle Recognition using Rule Based Methods, 1987. Turing Institute Research Memorandum TIRM-87-018.
- [41] A. M.-C. So. Improved approximation bound for quadratic optimization problems with orthogonality constraints. *Symposium on Discrete Algorithms*, 2009.
- [42] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, New York, NY, 2002.
- [43] J. W. Tukey. A Survey of Sampling from Contaminated Distributions. In I. Olkin, editor, *Contributions to probability and statistics: essays in honor of Harold Hotelling*, pages 448–474. Stanford University Press, Stanford, CA, 1960.
- [44] G. Watson. Characterization of the Subdifferential of Some Matrix Norms. *Linear Algebra Appl.*, 170:33–45, June 1992.
- [45] H. Xu, C. Caramanis, and S. Mannor. Principal component analysis with contaminated data: The high dimensional case. *preprint*, pages 1–37, 2009. arXiv:1002.4658.
- [46] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via Outlier Pursuit. *preprint*, pages 1–24, 2010. arXiv:1010.4237.
- [47] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *NIPS 23*, pages 2496–2504. 2010.